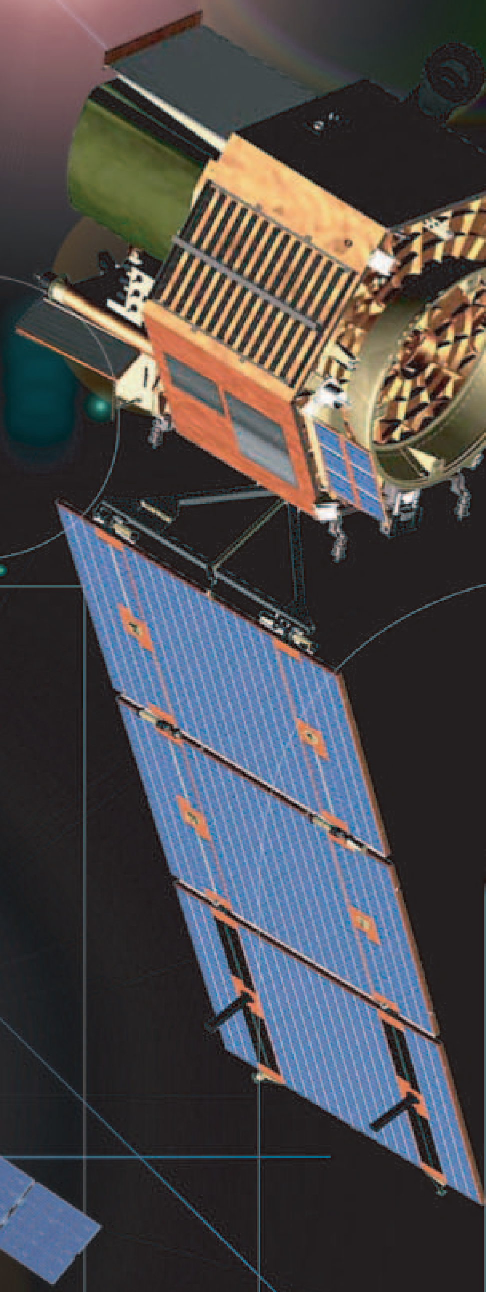
National Aeronautics and Space Administration

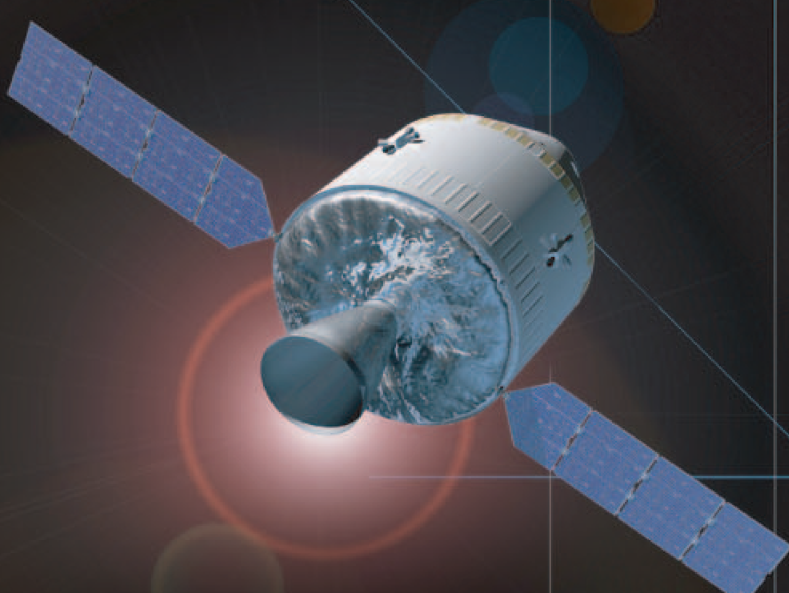
2005-2006

Intelligence Report

Exploration Technologies

Enabling New Crewed and Robotic Missions for NASA





NASA Ames Research Center

Intelligent Systems Division

Exploration Research

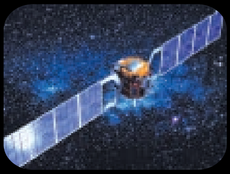
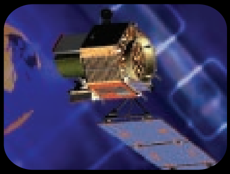
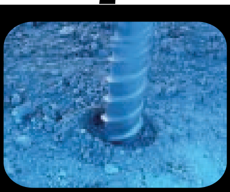
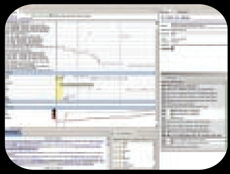
NASA Ames Research Center leads the agency in the development of advanced software and information technology capabilities and research for Exploration, Science, and Aeronautics. We perform mission-driven research and development to enable new system functionality, reduce risk, and enhance the capability for NASA missions.

NASA's focus has clarified around Exploration, and the Intelligent Systems Division's expertise and capabilities are being called upon to support NASA's missions. The Crew Exploration Vehicle (CEV) and teams of humans and robots working in space will all require advances in integrated systems health management, autonomous systems, mission operations, and reliable software.

Exploration requires the best of NASA's technical community to step up to providing the technologies, engineering, and systems to regain the frontiers of the Moon, and to extend our reach to Mars and beyond. Our technologies will be implemented in in the CEV, Crew Launch Vehicle (CLV), and robotic missions; embedded in operations; flown on spacecraft and aircraft; and used by astronauts. We will lead the development of the Intelligent System technologies to enable Exploration for all of NASA.

Dr. David Korsmeyer
Chief, Intelligent Systems Division
NASA Ames Research Center

Table of Contents


Research Areas

The Intelligent Systems Division is dedicated to mission-driven research: basic and applied research driven by clear problems that must be addressed to enable future NASA mission capabilities. The typical focus of division work is on intelligent, adaptive, and autonomous systems for sustained operations and operation in complex environments or under emergency conditions.

Robust Software Engineering

NASA exploration missions to the Moon, Mars, and beyond will increasingly rely on highly capable software to achieve greater levels of autonomy and robustness. Managing the risk in an affordable manner is one of the greatest challenges for the space agency.

The goal of automated software engineering is to increase software quality, reliability, and productivity. The cross-cutting research done by the Robust Software Engineering group draws upon several disciplines including Artificial Intelligence (particularly automated reasoning and knowledge representation), Formal Methods, programming language theory, mathematical logic, and advanced compiler methods. Research is done in the context of NASA applications, aiming to make large impacts rather than incremental advances.

Contact Michael Lowry: Michael.R.Lowry@nasa.gov

Collaborative and Assistant Systems

As NASA missions become longer and more scientifically complex, so will the ongoing participation of and cooperation between individuals in many different locations. At the same time, human/machine interactions will become increasingly complex.

The goal of the collaborative and assistant systems research is to design new information technologies and collaboration tools that facilitate the process by which NASA engineers, scientists, and mission personnel collaborate in their unique work settings. The research activities in this area focus on applying information management, artificial intelligence, and computer-supported cooperative systems that are more usable, that augment the human cognition, and that facilitate the specialized work of distributed teams in NASA mission settings.

Contact Ernie Smith: ernest.e.smith@nasa.gov

Discovery and Systems Health

The Discovery and Systems Health (DaSH) technical area focuses on challenges in understanding engineering and science data. The engineering data understanding work is centered around the emerging systems engineering discipline of Integrated Systems Health Management (ISHM). Ames is NASA's premier ISHM research and development facility, with strengths in design of health management systems, ISHM systems engineering, sensor selection and optimization, monitoring, data analysis, prognostics, diagnostics, failure recovery, diagnostic decision aids, data and knowledge management, and ISHM human factors.

Scientific data understanding work targets large-scale data analysis problems in data-rich domains such as earth science and cosmology. In addition, DaSH is involved in data analysis and mining for a variety of other NASA missions including aviation safety and security, the Space Shuttle program, and the NASA Engineering Safety Center. Resident expertise includes machine learning, data mining, text mining, statistical pattern recognition, and exploratory data analysis.

Contact Serdar Uckun: Serdar.Uckun@nasa.gov

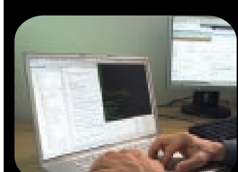
Autonomous Systems and Robotics

The Exploration Vision calls for closer cooperation between humans and robots than ever before. Creating robust robotic assistants, as well as making key spacecraft systems self sufficient, requires building systems that can adapt their behavior to environments that are complex, rapidly changing, and incompletely understood. Ames Research Center has unique expertise and agency leadership in applying autonomy to NASA missions, developing the individual technologies required, and integrating these pieces into autonomous systems for flight missions and terrestrial demonstrations.

Areas of research and development include adaptive control technologies, control agent architectures, embedded decision systems, evolvable systems, intelligent robotics, adjustable autonomy, distributed and multi-agent systems, goal-level commanding, and planning and scheduling.

Contact John A. Schreiner: John.A.Schreiner@nasa.gov

10



ROBUST SOFTWARE ENGINEERING Software design and operation for complex spacecraft and space missions

11

V&V Verification and validation of neural network adaptive control software

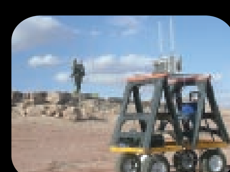
JAVA PATHFINDER Model checking to identify and eliminate software errors

12

CLARISSA Spoken dialog system for hands-free procedure execution

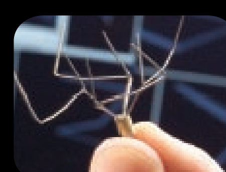
SUBVOCAL SPEECH RECOGNITION Facilitating noise-immune communication

13



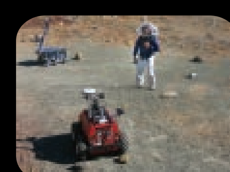
MOBILE AGENTS EVA support from software running on moving computers

14

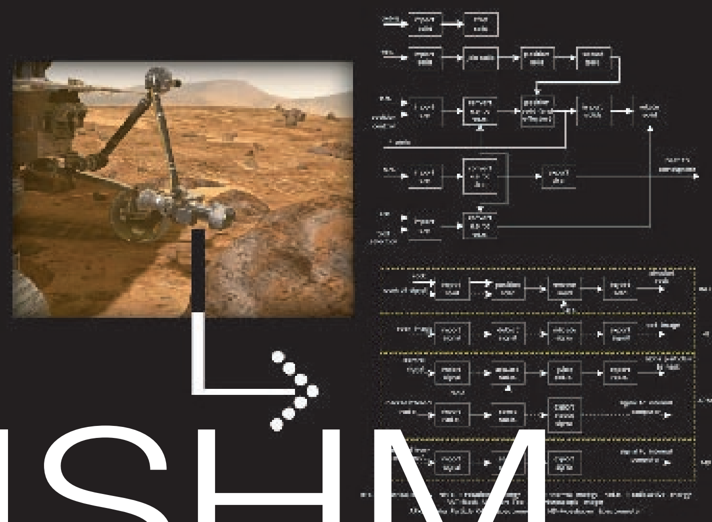


EVOLVABLE SYSTEMS TECHNOLOGY Evolutionary algorithms for system design and optimization

15

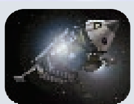


COLLABORATIVE DECISION SYSTEMS Developing self-reliant, cooperative human/ machine systems



ISHM

As NASA prepares to design and build the next generation of space transportation systems, there is a need for new systems health management technologies.



detect faults

correct faults

Integrated Systems Health Management

Today's space missions depend on small armies of flight engineers and ground controllers who must analyze enormous amounts of data to monitor and control the operation of the system. This is laborious and time consuming. Sometimes, particularly during the critical minutes of launch and ascent, decisions such as whether to abort the ascent or order crew escape procedures must be made in seconds. With deep-space robotic spacecraft the issue is not split-second immediacy but rather long communication delays; crucial decisions regarding spacecraft operations cannot wait for the round-trip transit time it takes for data to be returned to Earth, a decision made, and instructions sent back to the spacecraft.

Recent mission anomalies on the Space Shuttle, Mars Polar Lander, and Spirit rover have highlighted the need for improved integrated systems health management (ISHM) tools. Mission controllers and on-board crews need tools with improved situational awareness, on-board diagnosis, and recommendations for remediation and repair. However, it is not possible to simply list all possible situations that could arise during a mission and program computers with the appropriate responses. The sheer number of possible causes and outcomes—particularly when dealing with the interactions of multiple systems like propulsion, life support, avionics, and power generation—makes such an approach impossible. Future space systems will need built-in intelligence to detect faults, report them coherently, and manage them to assure safety and mission success.

Intelligent reasoning about faults requires models of subsystems and their interactions. The NASA Ames ISHM approach combines sensors with hardware and behavior models to track the system state and detect performance deviations. The models are also useful in system design, automated software development and validation, and mission simulation. Models can sometimes be learned automatically from flight hardware data. Ames also has experience with model development where failure data are rare, costly, or unobtainable.

NASA's Livingstone diagnostic system on Deep Space One was very successful, though it could not reason about continuous dynamics or multiple faults. Its successor, Livingstone 2, is now operating aboard the Earth Observing One spacecraft, setting new records for in-space monitoring and diagnostic performance.

Ames and the Jet Propulsion Laboratory recently tested their Inductive Monitoring System and Beacon-based Exception Analysis for Multimissions on Dryden Flight Research Center's F/A-18 flight research aircraft. The F/A-18's data bus and electrical, thermal, and vibration environments are similar to those of spacecraft propulsion systems, enabling the F/A-18 to serve as a low-cost ISHM testbed. Other Ames efforts include ISHM collaborations with Northrop Grumman and Honeywell, and a collaboration with Pratt & Whitney Rocketdyne on Space Shuttle Main Engine prognostics. Ames researchers are also developing fundamental technologies for autonomy applications such as rovers and drills for subsurface sampling.

ISHM technologies add value in vehicle and systems operations, in maintenance and logistics, and as part of the design cycle. Intelligent ISHM systems will make future spacecraft safer, less costly to maintain, and better able to sustain human presence in the harsh, unforgiving, and unpredictable environments of space, the Moon, and the solar system. Other potential applications arise anywhere that intelligent monitoring is cheaper than catastrophic failure—perhaps even extending to predicting ecosystem disasters on Earth.

Contact *Serdar Uckun - Serdar.Uckun@nasa.gov*

EUROPA

EUROPA: Enabling Spacecraft Autonomy

As spacecraft and systems become more complex, on-board intelligence is needed for control, system fault identification, and responses to unanticipated conditions. The Extensible Universal Remote Operations Planning Architecture (EUROPA) is an automated decision-making framework developed at NASA Ames that supports the construction of automated planning and scheduling applications.

EUROPA traces its roots to the successful Remote Agent Experiment (RAX) aboard Deep Space One. Later, it was integrated with JPL's APGEN activity plan editor to form MAPGEN, a tool used for scheduling daily science activities for the MER rovers. It has also been used in a number of development projects, including a contingent planner for K9 rover operations and a general control-agent architecture called IDEA.

EUROPA uses models of spacecraft systems, environments, behaviors, and goals to construct safe plans. This capability can be utilized in automated spacecraft and systems, and in interactive or automated planners for ground operations, observatories, remote sensing platforms, rovers, and other applications. The latest version, EUROPA 2, provides a component-based software library for reasoning about plans and schedules, and has been extended to handle complex continuous resources. It also offers improved performance, expressivity, reasoning capabilities, flexibility, extensibility, and modularity.

Formal models for representation and reasoning about actions and their consequences will be an essential part of future intelligent systems for space missions, linking mission planning with on-board reasoning, execution, and fault recovery. This will enable complex missions in unknown environments far from real-time human interaction, and will reduce the time and cost of planning missions.

Contact *Ari Jonsson - jonsson@ptolemy.arc.nasa.gov*

ADIS

Aviation Data Integration System

ADIS is a data repository and associated middleware that provides rapid and secure access to integrated digital flight data, weather data, airport operating condition reports, radar data, runway visual range data, and navigational charts. Previously, airlines' Flight Operation Quality Assurance programs de-identified all flight data by removing time, date, and flight number before performing any analysis of that data. This made correlation with other types of aviation data virtually impossible.

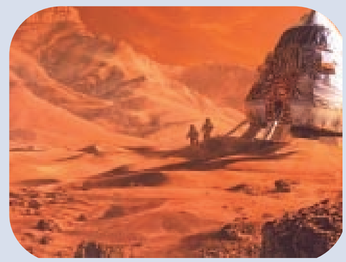
ADIS resolved this problem by using asymmetric key encryption to secure flight data identification information, thereby enabling aviation data to be associated with individual flights while satisfying airlines' concerns about privacy, data security, and sensitivity.

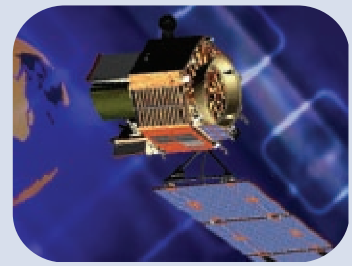
In March 2004 the FAA provided permanent funding to host the ADIS web server, making it permanently available to all airline Aviation Safety Action Programs in the country. In July 2004 the ADIS software was released to SAGEM Inc., which incorporated ADIS into AGS, their flagship flight data analysis software, distributed to about 85 airlines. This major NASA technology transfer achievement has the potential for greatly enhancing commercial aviation safety. NASA recognized this accomplishment by awarding the ADIS team a Space Act Award in November 2004.

Contact *Deepak Kulkarni - Deepak.Kulkarni@nasa.gov*

Livingstone

Software that can automatically detect errors in a space vehicle's systems and subsystems before complex problems become critical could significantly reduce mission operations costs; boost mission efficiency; and perhaps one day save a rover, a spacecraft, or even a human life.





Livingstone on Earth Observing One

Complex space systems will use on-board health maintenance software to track system state, detect faults, diagnose the causes, and recommend or implement recovery steps. Autonomous action before problems become critical could save a rover, a spacecraft, or even human life. By reconfiguring, a damaged spacecraft may recover functionality and continue to meet its mission goals.

Model-based integrated vehicle (or system) health management (IVHM/ISHM) uses sensor data plus a stored model of spacecraft components and functions. The same models can be used for mission simulation, software verification/validation, and autonomous or interactive planning and scheduling. The Livingstone fault diagnosis and recovery kernel from NASA Ames was a seminal technology. It flew on Deep Space One in 1999, teamed with an HSTS/EUROPA model-based planning and execution system. The success of this Remote Agent Experiment (RAX) has been called one of the ten greatest achievements of artificial intelligence.

Mission managers took an interest in Livingstone for In Situ Propellant Production, TechSat-21, and Advanced Life Support System projects. To meet their needs, researchers developed a new generation called Livingtone 2 (L2), incorporating temporal trajectory tracking (following multiple hypotheses, with backtracking and belief revision in light of any new evidence), diagnosing multiple faults across subsystems, and estimating probabilities of possible root causes. Early work focused on monitoring a high-fidelity simulation of the main propulsion system of the X-34 reusable launch vehicle and the actuators for the X-37 reusable reentry vehicle, the International Space Station Command and Data Handling bus, and a BabyBed re-implementation of RAX. Engineers also developed hardware/software models of several robotic probes, plus rapid-prototyping tools for high-fidelity simulators.

In 2004, L2 was uploaded to the Earth Observing One (EO-1) remote sensing satellite (launched by Goddard Space Flight Center in 2000). Ames also supplied models of the EO-1 spacecraft, including its instruments, cameras, and one of its processors. L2's function is to monitor and diagnose EO-1's imaging instruments and solid-state recorder under command of the Autonomous Sciencecraft Experiment and its CASPER planner from the Jet Propulsion Laboratory. L2 performance has exceeded expectations, with no software faults. L2 has detected and identified all simulated faults in the EO-1 imaging sequence, and is now demonstrating extended operations in space, with a longest contiguous run of 55 days and cumulative operations time of at least 143 days. This is a new record for model-based diagnostic software in space. With prolonged use, L2 may diagnose actual faults as they occur. It can even monitor spacecraft state and diagnose faults during transients, while the physical dynamics of the system are settling out. Such intelligent fault management could enable cluster operations and other challenging space missions.

Livingstone has proven valuable as a research tool in academia and as a diagnostic application at several NASA centers, and has developed a strong following in the aerospace industry through NASA's Space Launch Initiative. Continued development of such tools will make future NASA missions safer, more affordable, and more effective.

Contact *Sandra Hayden - shayden@email.arc.nasa.gov*

Autonomous Drilling

Autonomous Drilling

What do Spain's Rio Tinto, Devon Island in the high Arctic, and Gusev Crater on Mars have in common? All of these places are helping NASA researchers prepare to search for life on other planets.

On Earth, water is the key ingredient for life, and the Spirit and Opportunity rovers were sent to Mars to "follow the water." Now that they have found convincing evidence of past water in two very different locations on opposite sides of the planet, the next step is to follow the water trail beneath the surface—no easy task when this requires robotic drilling through the martian permafrost. This is where terrestrial analog environments come in—locations on Earth with extreme conditions similar to those that researchers believe they will encounter on extraterrestrial missions. These environments are where the needed automation and control technologies can be tested.

DAME

Drilling Autonomy for Mars Exploration (DAME), conducted by the Ames Intelligent Systems Division, is NASA's only project solely devoted to developing controls and automation software for extraterrestrial drilling automation. DAME uses a modified deep drill from Honeybee Robotics, the company that developed the Mars Exploration Rover (MER) Rock Abrasion Tool used by Spirit and Opportunity to grind off the surface of rocks so freshly exposed surfaces could be examined. The DAME team conducted its first field tests on Devon Island in August 2004. Drilling several meters into the frozen soil and permafrost, they produced a variety of fault conditions, along with the data needed to identify and analyze them.

In July 2005 the team returned with a full-scale Mars-prototype Honeybee deep drill to test the fault diagnosis and artificial intelligence software developed at Ames. The drill was tested in the regolith-like fallback breccia inside Haughton Crater, drilling 2.1 meters into ice layers and permafrost similar to what one might expect to find near the surface in martian polar regions. The current drill automation architecture includes downhole diagnosis of different rock and ice strata, bit wear detection, and dynamic replanning capabilities when unexpected failures or drilling conditions are discovered. Major faults detected by the DAME software included bit choking, impingement, hard material/worn bit, and cuttings removal clog.

MARTE

NASA's most ambitious Mars analog project is the Mars Analog Research and Technology Experiment (MARTE), a joint effort between the NASA Ames Astrobiology Institute and the Spanish Centro de Astrobiologa (Center for Astrobiology). MARTE is a month-long simulation that combines long-distance communications, remote-controlled automated drilling, and life detection experiments. Rio Tinto's highly acidic chemical composition and sulfide mineral deposits are similar to those that scientists say might be found in the martian subsurface.

MARTE's 2004 field season focused on setup and testing of communication links for robotic drilling, and retrieval of core samples for analysis. The Honeybee Robotics MARTE drill is a highly automated 10-axis deep drill and core retrieval system, designed for subsurface sample recovery and hand-off from depths of up to 10 meters.

The 2005 field season added interpretation of drill mission results by remote science teams in a blind test. Full simulation requires three interrelated levels of automation, remote operations, and remote data acquisition software.

- *The remote operations subsystem comprises the computing hardware and software to allow remote operational control of the equipment at the borehole.*
- *A mission control center relays instructions to and receives data from the drilling site.*
- *Science operations centers in Spain and California work under the time delay and data transfer constraints they would encounter in an actual martian drilling mission.*

Advanced drilling technology also has other applications to NASA's Vision for Space Exploration. Extended visits to or habitation of the Moon, and eventually of Mars, will require at least partially automated construction of habitats in these environments, utilizing raw materials as much as possible.

Contact *Brian Glass - Brian.Glass@nasa.gov*

SOFIA AFP

Automated Flight Planner for SOFIA

SOFIA, the Stratospheric Observatory for Infrared Astronomy, is NASA's next-generation airborne astronomical observatory—the world's largest, making observations that are impossible for ground-based telescopes. A joint project with the German Aerospace Center (DLR), SOFIA consists of a 747-SP aircraft carrying a 2.5-meter telescope and other instruments for sensitive measurements of astronomical objects over a wide wavelength range (0.3 microns to 1.6 mm). Airborne telescopes offer reduced line-of-sight water vapor compared to even the highest ground-based telescopes, which is crucial for infrared astronomy. At the same time, airborne telescopes last longer than most space-based telescopes, and offer greater flexibility to take advantage of new instruments, software, and hardware advances. SOFIA will fly at altitudes ranging up to 45,000 feet, two to three nights a week for at least twenty years. Studies will include planets, comets, and asteroids; interstellar cloud physics; star birth and death; planet formation; nebulae, dust, and molecules in galaxies; black holes; and ultra-luminous infrared galaxies.

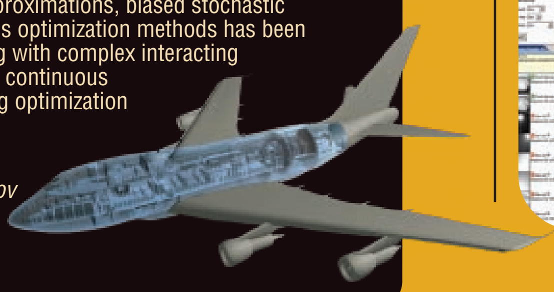
SOFIA operations scheduling is extremely complex. Flight planning requires choosing a set of observations to perform over the course of several flight days, during which a single instrument is installed on the aircraft. Typically, there will be too many projects to observe in any given period. This means that a subset of observations must be selected according to priority. The scheduling process involves a trade-off between the priority of individual projects and what is possible to observe. On average, SOFIA can support 1-5 projects (5-15 observations) per flight, and schedules will last between 2 weeks and 2 months.

Complex constraints govern the planning of even a single flight. Instrument observations may require a specific time of day, altitude, telescope elevation angle, and duration. Time of day (for high angle of observation) can sometimes be traded against both position and altitude in order to minimize line-of-sight water vapor. Motion of the aircraft itself is governed by differential equations, whose solutions are in turn subject to non-linear constraints, creating a nonlinear optimization problem. There are also Special Use Airspace constraints, wind and weather constraints, and costs related to flight duration and fuel load. Changing any scheduled observation may affect the feasibility of other observations, and may introduce dead legs (with no observations) in the flight plan.

Intelligent Systems Division researchers have developed an Automated Flight Planner (AFP) to plan SOFIA flights with minimal human intervention. Analysis of a trade study using 50 hours of requested observations over eight days identified operational schedules that could increase scheduled observations by 1/3, decrease line-of-sight water vapor by 1/3, or decrease fuel consumption by 60,000 lb (or 10,000 gallons) per flight. At present fuel costs, this is a potential cost savings of $30,000 per flight; with SOFIA expected to fly roughly 140 flights a year, this translates to a potential cost savings of $4.2M a year. Automation enables more rapid generation of the trade space than is possible with human operators, permitting effective decisions during SOFIA operations.

AFP techniques for finding good flight plans could also find use in reconnaissance, environmental monitoring, or disaster relief, and in other complex planning and scheduling domains. The integration of computational techniques such as heuristic search, well-founded approximations, biased stochastic sampling, and continuous optimization methods has been shown capable of dealing with complex interacting constraints, discrete and continuous decisions, and competing optimization criteria.

Contact *Jeremy Frank - Jeremy.D.Frank@nasa.gov*



Mission Tools

Mission Operations Tools

Mission operations is the process by which mission decision makers and spacecraft operators coordinate many data sources and software systems to determine the current state of the spacecraft and its environment; negotiate which desired activities will actually be performed next on board the spacecraft; generate and validate the spacecraft commands that will perform the selected activities; and receive, organize, and distribute the resulting data products from the spacecraft. Tasks in this process can range from determining if a desired image was successfully generated and exists in the ground data system to verifying that a set of commands to be uplinked to the spacecraft implements the desired effect without violating any spacecraft resource limits or flight safety rules.

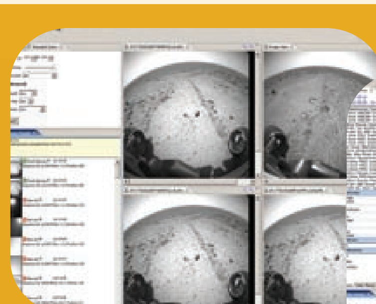
Automated and mixed-initiative (human/automated) decision support tools from Ames Research Center have been critical to the success of the Mars Exploration Rover (MER) mission. Without the help of these tools it would have been difficult or impossible for mission planners to cope with the massive amounts of science and engineering data required daily in their work.

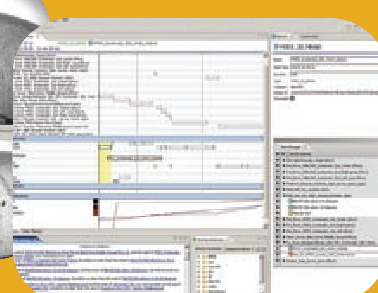
But one serious limitation of the MER tools is that they were developed separately rather than in an integrated fashion. To use a combination of these tools on a mission requires interfaces between systems, and the export or translation of data. Building on their MER experience and successes, researchers at Ames, in collaboration with scientists and engineers from other NASA centers, are developing the next generation of mission tools, with a focus on efficiency, flexibility, and interoperability.

Ensemble

The aptly named Ensemble, a joint Ames/Jet Propulsion Laboratory (JPL) project, is a cross-center platform for the development, integration, and deployment of mission operations software. Based on the popular open source Eclipse application platform, Ensemble is designed around a set of mission-specific and tool-specific plug-ins, and can be reconfigured or upgraded as required during the course of a mission. Each application is assembled from different subsets of the Ensemble components, but all use the same data representation. This allows science planning, modeling, activity planning, plan representation, and command sequencing to work together seamlessly.

Scientists can browse images and create targets from Ensemble's downlink or data browsing perspective. In the planning perspective activities can be arranged into plans, edited, and linked together with





mission or hardware constraints. A combined view allows planners to mix and match components (for instance, to show the details of an activity next to an image illustrating the target of the activity).

A growing set of Ames and JPL tools is available as Ensemble plug-ins.

- **Ames Ensemble developers are assisting their JPL collaborators to re-deliver the MER Science Activity Planning tool as Ensemble components.**
- **The 2007 Phoenix Lander Science Interface tool is based on Ensemble, using components from Ames and JPL.**
- **Ensemble is also the baseline for the 2009 Mars Science Lander's activity planning function.**

Contact *James Kurien - James.A.Kurien@nasa.gov*

Mission Control Technologies

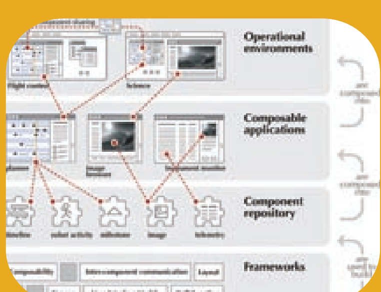
A successful collaborative distributed mission needs more than just tools that play well together. It also requires frameworks and interfaces that facilitate such interoperability. The Mission Control Technologies (MCT) project is developing software technologies and design methods for modular, evolvable, distributed ground data systems and mission operation systems to meet those needs.

In current mission operations systems, each application serves a user base in a specific discipline or functional role. To support its tasks, the application embodies specialized functional knowledge and has its own data storage, data models, programmatic interfaces, user interfaces, and customized logic. By contrast, MCT takes a modular approach. User-configurable collections of objects and services share an information model with consistent interfaces and interactions.

Each component of the MCT architecture is built to address specific mission issues, and collections of composable distributed objects can be flexibly assembled based on mission requirements and mission operations processes. Development standards and user interactions are consistent through contractually specified interfaces, with relationships and rules captured in information models, not as properties of the objects. This allows different organizations to create interoperable components without knowing in advance which components will be working together. It also enables reuse, migration, and integration of components and capabilities onboard and on the ground

MCT users can build custom environments to fit their individual needs and ways of working. MCT's services allow distributed access to objects by any software component or application. All components remain autonomous and independent, and every representation of data is tied to an underlying object. For instance, the same project plan can be opened as a timeline, a list, or in both representations simultaneously. Since anything from mission tools to specific components to individual data units can be an interoperable component, the MCT framework permits much finer-grained levels of granularity than those built into the tools or other components themselves.

Contact *Jay Trimble - Jay.P.Trimble@nasa.gov*



MER

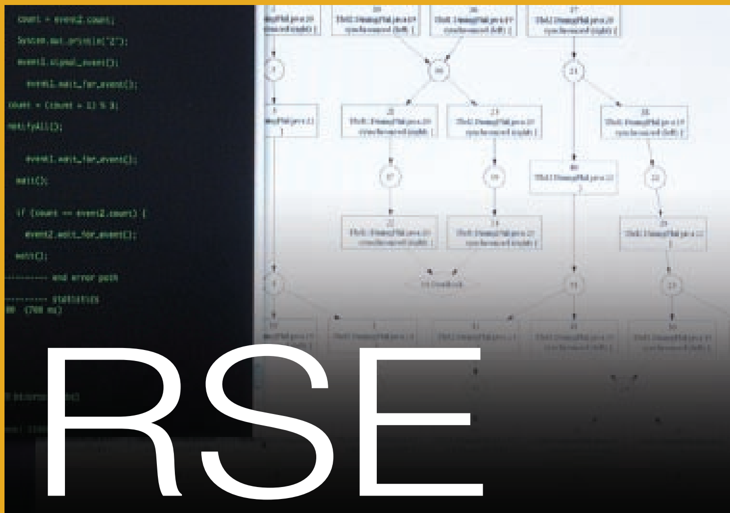
In January 2004 the rovers Spirit and Opportunity landed on Mars for what was planned as a 90-day mission. Now in the third extension of that mission, they are still exploring and sending back valuable science data. Mission Operations remains at the Jet Propulsion Laboratory in California, but participating scientists have returned home and are now collaborating remotely, as they will in future long-duration NASA missions.

Ames-developed mission support tools continue to make this distributed operation possible. The Collaborative Information Portal (CIP) is a suite of scheduling and data management tools designed to facilitate long-distance collaboration. Each sol, or martian day, mission scientists and engineers analyze the data arriving from Mars and begin planning the next actions of the rovers. Part database, part visualization software, CIP enables scientists around the world to inspect incoming data from the rovers, retrieve images, science and engineering data products, reports, plans, post comments, and coordinate schedules in all mission time systems.

Other tools such as the MERBoard (a large touchscreen-based collaborative computer), and VIZ (a tool which produces a photorealistic 3-D representation of the rover worksite) further aid in the process of collaborative data understanding, analysis, communication, and enhanced situational awareness. MAPGEN (Mixed Initiative Activity Planning Generator) is used to validate the daily rover plans, where each target/activity combination specifies a location and a task for a rover. MAPGEN automatically takes into account hundreds of flight rules and resource constraints as it generates a plan including as many requests from the scientists as possible. MAPGEN has helped MER science teams build plans for the rovers that on their most difficult scheduling days enabled up to 50 percent more science activity to take place.

Contact John Schreiner-John.A.Schreiner@nasa.gov

Update



As mission requirements call for more sophisticated capabilities like autonomy, systems health maintenance, and crew-centered operations, so too does the size and complexity of the necessary software.

Robust Software Engineering

Software has an increasingly critical role in the design and operation of new spacecraft and space systems. As mission requirements call for more sophisticated capabilities, such as autonomous systems, integrated systems health maintenance, and crew-centered operations, the need for even more complex software increases. The software requirements for the new Crew Exploration Vehicle to be built as part of NASA's Project Constellation will greatly exceed those of Project Apollo, the Space Shuttle, and even the International Space Station. And with increased software requirements will come increased criticality and risk. It is therefore essential that software engineering tools and techniques be developed that can enable the development of software that is affordable, reliable, and sustainable, while meeting these new mission requirements.

The sheer size of the software programs that will be required for Project Constellation poses a challenge. Over the past four decades of space exploration the increase in size of flight software has been nearly exponential, ranging from the Voyager and Galileo missions of the 1970s and 1980s with less than ten thousand lines of code to the Shuttle with nearly 500,000 lines of code and the International Space Station with 1.7 million lines of code. Small armies of programmers are required to write, debug, and certify this software for human-rated flight.

Fortunately, current software engineering research is developing techniques that can deal with such challenges, and researchers in the Intelligent Systems division at NASA Ames are recognized leaders in their field. To date, they have developed several innovative solutions to the challenges posed by complex software verification and validation:

- The difficult task of verifying large software systems can be addressed by compositional verification, which takes a "divide and conquer" approach that breaks up the verification of a system into smaller tasks that involve the verification of its components. This approach has recently been used to automatically detect problems related to redundancy management and attitude control handover between the US and Russian sides of the International Space Station.

- Complex software containing loops is extremely difficult to verify, as loops may introduce infinite program executions. A technique called loop invariant generation now provides a framework for proving properties of looping programs, making it possible to automatically prove complex properties of programs.

- System-level verification automatically decomposes global (or system-level) requirements into local properties, which are cheaper—in terms of both time and consumed memory—to check. Using this technique, division researchers developed and applied automated verification and validation to the K9 rover executive (containing 35,000 lines of code) at different stages of software development. This created an exhaustive analysis of design models of the executive, and created a comprehensive set of requirements (both English and formal descriptions) for key concurrency and plan execution properties. Design models and requirements can be successfully reused for design and analysis of future advanced executives, which had a direct impact on new executive design: Based on analysis results, the developer created a new executive, with simplified architecture to increase modularity and to facilitate reuse.

Working with software developed and deployed on missions by their division colleagues, the Robust Software Engineering team is continually developing and refining sophisticated techniques for finding previously undetected bugs and other software errors. Tools using these techniques have been applied to mission software from Deep Space One, Mars Pathfinder, the Space Shuttle, and the International Space Station, successfully verifying the correctness of the code in some cases, and locating difficult-to-detect faults in others. In addition, the team has been working with colleagues at Goddard Space Flight Center, the Jet Propulsion Laboratory, Johnson Space Center, Marshall Space Flight Center, and United Space Alliance to deploy tools and technologies for formal inspections, code analysis, and defect classification on many mission applications, including International Space Station payloads and spacecraft flight software. Several of these efforts have been undertaken as part of the NASA Software Engineering Initiative, led by the Office of the Chief Engineer. The techniques used are at the cutting edge of current software verification and validation research, and these efforts are already being directed to the software engineering challenges faced by Project Constellation.

Contact *Michael Lowry - Michael.R.Lowry@nasa.gov*

V&V of Adaptive Flight Control Software

Reliable, adaptive control systems will be needed to fulfill the mission requirements of future aeronautics and space missions. Adaptive control systems can be designed to retain robust control in situations involving damage to the vehicle that would otherwise lead to catastrophic failure. But a serious difficulty hindering the deployment of adaptive software technology in the national air space is the requirement to prove that such systems can operate with very high reliability.

In an effort to study software performance and reliability, NASA Ames and Dryden, Boeing Phantom Works, and the Institute for Scientific Research Inc. have teamed up to test a neural adaptive flight control system in 2006. The flight control system will be tested on a specially modified F-15 aircraft. Unlike normal F-15s, this testbed has large canards to simulate aircraft damage. As the canards are deflected, the airflow over the normal wing control surfaces is disturbed, thereby simulating aircraft damage.

The F-15 is also special in that it has two flight control systems: a normal F-15 flight control system, and a neural controller designed to help the pilot fly the aircraft during simulated damage conditions. The use of a dual control system allows the adaptive control software to be tested without endangering the safety of the aircraft.

The flight test will evaluate a new software tool called the confidence tool. This tool was developed at NASA Ames to offer a means to assess neural network reliability in flight. The confidence tool monitors the neural work and system outputs to determine a statistical measure of system reliability. From this dynamic measure, it can be determined if the controller is still within the safety margin, even under unanticipated changes to the flight environment.

A signal of low-control performance can be used for early fault detection analysis, thereby providing anticipatory warning of incipient software reliability problems. The development of such capability will hopefully pave the way for FAA certification of adaptive control software technology.

Contact *Stephen Jacklin - Stephen.A.Jacklin@nasa.gov*

JAVA Pathfinder

JavaPathfinder

JavaPathfinder (JPF) is part of an effort to develop tools and methods to identify and eliminate software errors in NASA's increasingly complex and mission-critical software systems. Java Pathfinder uses a technique called model checking that allows all possible executions of a Java program to be analyzed in order to find errors. JPF tests Java programs by running them through a series of trials to look for conditions under which they will fail. If an error is found, the software checker reports the entire process that leads to the bug. Typical errors being targeted include deadlocks and mutual exclusion and assertion violations.

As an example of this capability, JPF was used to detect inconsistencies in the executive software for the K9 Rover at NASA Ames. Elements of JPF were also used to develop an advanced verification tool for Livingstone 2, which is a model-based diagnosis system currently flying on the Earth Observing One spacecraft and an example of the kind of autonomy software that will be crucial to future NASA missions.

The Java PathFinder project was initiated after an experiment in 1997 in which part of the Remote Agent, an artificial intelligence-based software component of the Deep Space One spacecraft, was analyzed by hand, translating part of the Remote Agent code into the language of an existing model checking tool. The analysis identified five classical multi-threading errors that had not been caught by normal testing. One of these errors was a situation where two threads executing in parallel interacted in an unexpected manner. The first version of Java PathFinder, JPF1, completed in August 1999, automated this process by translating from Java to the Spin model checker from Bell Labs. JPF1 demonstrated for the first time the feasibility of model checking Java source code directly without manual modification of the code. JPF1 was applied to two software systems developed at NASA: a satellite file exchange module developed at Goddard Space Center, and a ground control module for the Space Shuttle Launch facility at Kennedy Space Center. Limitations in the translation-based approach led to the development of Java PathFinder 2 (JPF2), which model checks Java bytecode directly using a custom Java Virtual Machine. Subsequently, advanced testing techniques were integrated into JPF2 and used to automatically find an intricate error that was known to be in the DEOS real-time operating system developed by Honeywell for business aircraft.

A new version of Java Pathfinder, JPF3, is now freely available to developers under the NASA Open Source Agreement, an open-source license approved by the non-profit Open Source Initiative. By incorporating a number of changes to enable developers to customize the way that JPF searches and observes the program being tested, the open-source version of Java PathFinder will enable broad collaboration with universities and industry who are interested in using the tool. Java PathFinder was downloaded over 3,400 times in the first five days it was available on SourceForge.net.

While specific research projects are focused on the verification and validation of NASA software, the recent open-source distribution of JPF code enables the broader use and development of JPF capability for other NASA mission uses, and leverages the enthusiasm of the large number of developers in industry who are also working in this area.

Contact *John Penix - John.Penix@nasa.gov*

Clarissa

In environments where the most basic of tasks must follow strict procedures, efficiency and accuracy are vital to mission success. NASA's Clarissa spoken dialog system is designed to free astronauts' hands and eyes from paper and electronic procedures so they can work more efficiently and safely.

Today, an astronaut reads from paper procedures or a PDF viewer on a laptop computer. This requires the astronaut to shift attention from the task at hand to turn pages or scroll through computer screens. Clarissa can play the role of a procedure reader, as well as answer simple questions, display pictures, read ahead, take voice notes, and navigate through the procedure to other steps when commanded to do so.

The system was designed in an iterative development process, with input from astronauts, trainers, and procedure writers. It has a grammar-based vocabulary of about 260 words, and supports about 75 different commands. Application-specific spoken command grammars were constructed using the open source Regulus platform. Users can scroll forward or backward, move to an arbitrary new step, preview or read out a non-current step without losing their place in the procedure, open a subprocedure, and read safety-critical portions of the procedure in a mode which checks aggressively that steps have not been skipped. Other commands include support for recording, playing and deleting voice notes, setting timers and alarms, and querying status. Speech recognition is in "open mic" mode, using a noise-canceling microphone. Users can undo or correct any misrecognition or inappropriate response with a single voice command.

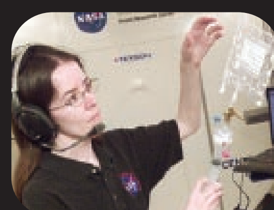
The current experimental version of Clarissa runs on a standard International Space Station (ISS) laptop, and browses procedures written in an XML format. Data files must be able to capture the exact content of the current written or on-screen procedure, adding any additional information necessary for verbal output. The markup language includes constructs for branch points, conditional steps, querying the user for numerical values, and linking to subprocedures. For the initial experiments, five procedures have been converted into Clarissa-compatible XML form: three for sample collection and testing of the Station's potable water supply, and two for space suit checkout.

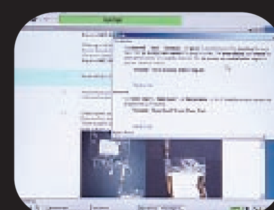
Clarissa software was installed on the ISS in January 2005. It was used by Expedition 11 Science Officer and Flight Engineer John Phillips on June 27, 2005 in the first-ever trial of a spoken dialogue system in space. During the test, Phillips successfully completed the interactive Clarissa training procedure, which exercises all the main system functionality. Speech recognition and dialogue management both appeared to function well. Plans call for Clarissa to be used while executing ISS water testing procedures during Expedition 12. Ultimately, the goal is for Clarissa to be able to read and navigate any ISS procedure as part of the International Procedure Viewer.

The most important improvement to the system since its deployment has been development, through a research collaboration with Xerox Grenoble, of an approach that lets the system more accurately distinguish between commands directed at it and side conversations, decreasing the error rate from 10 percent to about 5 percent.

Clarissa has a visual interface and accepts a limited set of keyboard commands. Future possibilities include using eye movements and touch screens for navigation, which can be integrated with dialogue. The tool paves the way for hands-free interaction with robotic assistants, including systems on robots that could help astronauts diagnose medical conditions and monitor life support. It could also be used for hands-free operation by ground-maintenance crews and mission controllers.

***Contact** Beth Ann Hockey - bahockey@email.arc.nasa.gov*





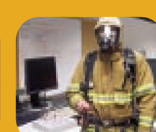
Subvocal Speech Recognition

Speaking is usually defined as communicating orally, making vocal sounds perceived by the auditory system to form words that express thoughts and ideas. But the physical production of speech involves more than the mouth and larynx. Whenever a person reads or thinks of a phrase, the brain sends speech signals to the tongue and vocal cords, even if the person is not speaking out loud or moving their lips or face. And, as NASA Ames researcher Chuck Jorgensen has found, this opens up all sorts of interesting possibilities, because these signals can be picked up and translated into recognizable symbols.

Some of the possibilities have direct relevance to NASA's space and aeronautics missions. For instance, subvocal speech could facilitate communication in noisy aircraft or spacecraft environments; by astronauts wearing spacesuits; and for the silent control of computers, rovers, and other robotic devices. But the technology lends itself equally well to many other medical, commercial, and public safety applications. Once the subvocal signals are detected and processed, many physical limitations of the human body that prevent sounds from being produced or adverse conditions in the environment that prevent them from being heard would become irrelevant to communication.

The keys to the system are its sensors, signal processing, and pattern recognition. Ames scientists use small, button-sized sensors stuck under the chin and on either side of the Adam's apple to capture the nerve signals in the throat. The technology sends the signals to a processor, then to a computer program that translates them into words or other commands, in a four-step process:

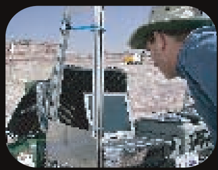
- **Data acquisition and signal segmentation**
- **Signal feature creation**
- **Classification classifier training, using neural networks**
- **Recognition, playback, and control**



In their first experiments, researchers taught the system to recognize words such as "stop," "go," "left," "right," "alpha" and "omega," and the numbers zero through nine. One demonstration controlled a graphic model of the Mars rover and another used a simple set of commands in a web browser. Since then, they've increased the system's vocabulary and begun teaching it how to detect the differences between vowel and consonant patterns, as a step toward full-scale subvocal recognition. They are also testing "non-contact" sensors that can read muscle signals through a layer of clothing.

Eventually, such "subvocal speech" systems could be used by astronauts in spacesuits; to control rovers and other vehicles; in noisy locations like airports, space habitats, or construction sites; by firefighters using pressurized breathing apparatus; or even as an adjunct in traditional voice-recognition programs to increase accuracy. People could also use the system for silent communication in public places, and those with physical impediments could use it to converse normally despite speech impediments or to control devices such as wheelchairs, prosthetic devices, and computers.

***Contact** Charles Jorgensen - Charles.Jorgensen@nasa.gov*



Mobile AGENTS
Architecture for EVA Support

Extra-vehicular activity (EVA) for surface exploration or in space has been risky, expensive, and highly dependent on ground operation support. Ames's Mobile Agents Architecture assists with EVA scheduling, navigation, equipment deployment, telemetry, health tracking, and field data collection and management. Computers supply or log routine information, alerting support crew during anomalous situations. This approach aims to eliminate manual telemetry readouts, reduce human monitoring by 80%, and automate 60% of dynamic EVA replanning.

The name Mobile Agents refers to software running on moving computers, communicating by a wireless network over a complex terrain of hills and canyons. Physical communications support is tied to a workflow toolkit, collaborative engineering methodology, voice-commanded data management, and various tools for field operations support. Simulation and visualization capabilities help with EVA planning and subsequent analysis. Such capabilities may someday support space construction, habitat maintenance, and other exploration tasks.

The distributed, open architecture integrates wireless biosensors, cameras, GPS, and robotic assistants. It proactively transmits data, helping to inform and coordinate EVA crew (and their autonomous robots), surface habitat crew, and remote support teams—essentially automating Apollo's CapCom (Capsule Commander) role. Software agents—incorporating models of activities and devices—interpret and transform data to improve human-robot-computer operations. For instance, agents monitor EVA progress and consumables, dynamically replanning EVAs and notifying crew of any needed changes or potential problems.

The software architecture (based on the Brahms agent-based modeling system) can link more than 60 agents, including six voice-commanded robots (Johnson Space Center's Boudreaux, Thibodeaux, and Scout, and Ames's PER, Gromit, and K9). It integrates six levels of hardware and software, down to customized interaction protocols for specific astronaut activities. It controls devices (including robots) or provides a hands-free interface to them, manages data flow and work flow, automates key supervisory functions, and supports automated alerting and advising. Through voice commanding, astronauts can store place names, photo locations, sample sites, and voice notes in a ScienceOrganizer database, with automated entry on TerraServer maps. The science database is also mirrored from the habitat to remote scientists and mission support.

The Mars Society's Desert Research Station (MDRS) Rotation 38 provided an excellent, cost-effective venue for Mobile Agents field testing. Teams from JSC and Ames participated in two intensive weeks of system integration, developing human-robot operations concepts and requirements while doing authentic geology exploration in a Mars analog environment. EVAs of more than two hours were supported, including at least a dozen activities planned on the same day by the crew (with advice from the remote science team's review of previous work). Astronauts interacting with the system worked for as long as 40 minutes without needing help from the support team. Such field tests permit ethnographic studies during competitive testing of alternative technologies and protocols.

Contact *William Clancey - William.J.Clancey@nasa.gov*

Evolvable Antenna

NASA Exploration Missions will require new advances in engineering design and optimization. Evolvable Systems technologies are poised to deliver these advances through the use of algorithms that can automatically design and optimize systems without being explicitly told how to do so.

Evolvable Systems Technology

Design of a new system can often be viewed—and implemented—as a search through choices in a design space. Candidate configurations are evaluated, possible improvements are explored, and optimized designs compete for selection. This is the approach of the Ames Evolvable Systems Group (ESG), which uses adaptive and evolutionary algorithms to solve engineering problems in novel ways.

Evolutionary algorithms—chiefly genetic algorithms, genetic programming, artificial neural networks, and simulated annealing—can design and optimize systems without being given explicit design rules. The problem domain must permit computational simulation and design evaluation. In situ evaluation is especially powerful, as when a spacecraft antenna's performance is affected by the physical structures and configuration of the spacecraft. ESG has developed antennas that exploit in situ reflections and interactions, achieving reduced device size, complexity, and cost while increasing performance and reliability.

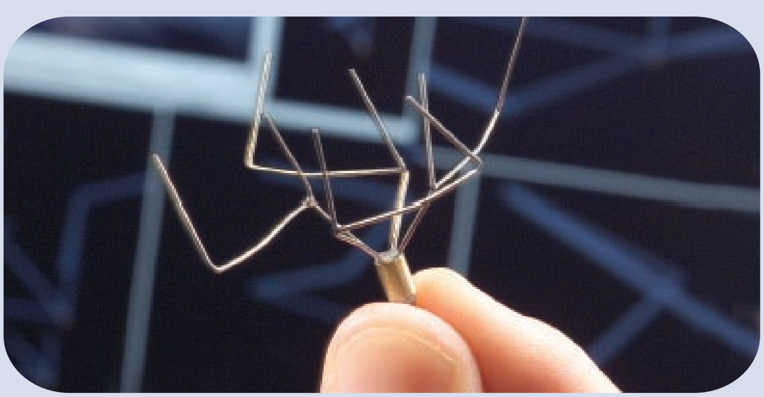
Two such antennas were recently developed. An X-band antenna for the New Millennium Space Technology 5 (ST5) mission—composed of three nanosats to study Earth's magnetosphere—will be the first evolved hardware flown by NASA. The circularly polarized antenna needed to combine wide beamwidth and wide bandwidth. ESG's evolved design was lighter, cheaper, and had fewer parts and higher bandwidth than a competing human-designed antenna. Its field pattern also allowed use over a wider range of elevation angles. Mission planners then changed the ST5 orbital altitude and functional requirements, and ESG produced a second design in less than four weeks.

ESG's JavaGenes—available as open source software—was used to evolve a crossed-Yagi phased array antenna for TDRSS-C (the next generation of Tracking and Data Relay Satellite Systems for communicating with the Shuttle, International Space Station, and Earth satellites). Phased array antennas can be pointed electronically, changing the beam direction with no moving parts. The study used ESG's Dareven cluster and several NASA Advanced Supercomputing computers—including Columbia—to optimize the antenna's boom length, spacing between crosses, and cross size.

ESG has also designed fault recovery for field-programmable gate arrays (FPGAs), which are often used for control functions aboard spacecraft, habitats, and autonomous rovers. If temperature or ionizing radiation damages an FPGA device, its real-time, on-board recovery system could automatically evolve a new wiring pattern for the faulty chip, restoring useful functioning.

JavaGenes has evolved pharmaceutical molecules; atomic force field parameters; digital and analog circuits; Earth Observing Satellite (EOS) schedules; and nanoscale structures such as micro-electromechanical system (MEMS) gyroscopes for precision navigation. Evolvable systems can also design large-scale structures such as launch vehicles, spacecraft, rovers, and habitat modules, and research in evolutionary algorithms for multiobjective optimization can help mission planners explore complex trade-off spaces. The field combines and is making contributions to computer science, electrical engineering, robotics, biology, mathematics, and aerospace engineering.

Contact *Jason Lohn - Jason.D.Lohn@nasa.gov*



Collaborative Decision Systems

Systems and sub-systems must become "smarter," and automated systems and the human crew will need to function as a well-integrated robust team.

Meeting NASA's goals for the Exploration Initiative will require the sustained cost-effective pursuit of increasingly complex missions in space and on planetary surfaces. Flight crews in exploration missions will need to be largely self reliant in routine operation, with ground support providing rapid expert consultation and ad hoc support. Creating and executing such control strategies requires machine intelligence, coupled tightly with human intelligence, to become an integral part of future flight systems for both crewed and robotic missions. This means that systems and sub-systems must become "smarter," and that these automated systems, and the human crew, will need to function as a well-integrated robust team.

The Collaborative Decision Systems (CDS) Project at NASA Ames Research Center is creating technology to make vehicles and other systems more intelligent and enable "networks" of cooperating robotic systems to be deployed that can work cooperatively to prepare landing sites, habitation, or resources, and to extend the reach of human explorers.

CDS scientists and engineers perform inter-disciplinary research and technology development integrated across multiple research groups, with multiple principal investigators, and integrating results from multiple NASA centers and other technology providers. CDS both advances the state of the art in relevant technical areas and brings together existing component technologies into milestone-focused field and laboratory tests designed to feed into other exploration programs.

The CDS Project breaks down the problem of creating self-reliant systems into three parts:

1) ***Automation and System Intelligence***
Software technology for "smart" systems, including both autonomy for robotics and software for automating tasks such as planning and scheduling

2) ***Human-Automation Interaction and Multi-agent Teaming***
Software technologies for integrating autonomous systems with human flight and ground crew members to form robust, self-reliant teams supporting efficient, seamless interaction between human and robotic participants

3) ***Health Management Technologies for Autonomous Operations***
Tools to address fault detection, identification, criticality assessment, and risk mitigation issues at the systems and subsystem levels, so crews can manage spacecraft health autonomously without dependence on Earth-based mission support staff

In late 2005, a demonstration using testbed rovers at Ames will show CDS capabilities including:

- **Flexible astronaut robot command cycles**
- **"Many-to-many" (multiple source and destination) robot commanding**
- **Robotic supervision of other robots**
- **Real-time EVA crew monitoring and advising**

The demonstration will also include previously demonstrated breakthroughs in rover operations, including single-cycle instrument and tool placement, in which the rover operates autonomously to place an instrument against a selected target in an unprepared or unstructured environment; and worksite navigation and planning, which involves precision navigation along with operations and route planning to multiple points for tasks distributed over a worksite.

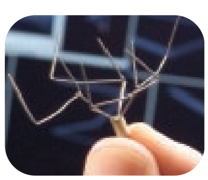
Contact *Robert Brummett - Robert.C.Brummett@nasa.gov*

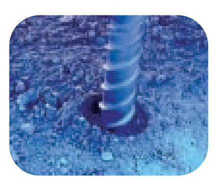
Intelligent Systems Division

Intelligence Report

Mail Stop 269-1

Moffett Field, CA 94035





http://ti.arc.nasa.gov